\textsc{\LaTeX}: the Multilingual Lion
\TeX{} meets Unicode and smart fonts

Jonathan Kew
SIL International
August 23, 2005
What is \TeX{}E?

- \TeX{} typesetting engine
  - including e-\TeX{} extensions
- Supporting the Unicode character set
  - inherently multilingual/multiscript typesetting system
  - greatly simplifies language support at macro level
- Using modern font technologies
  - TrueType, OpenType (all fonts supported by platform)
- With “smart rendering” support
  - Apple Advanced Typography
  - OpenType Layout features
  - for typographic features and complex scripts
Multilingual typesetting with \TeX

- Text input
  - escape sequences for non-ASCII characters
  - multiple 8-bit and double-byte codepages
  - use of active characters
  - preprocessors for complex scripts

- Font support
  - fonts limited to 256 glyphs
  - custom-encoded fonts with specific glyph sets
  - many different font encodings in use

- All tied together via complex \TeX macros
  - difficult to understand and extend
  - difficult to integrate with other packages
## Unicode support

### Traditional \TeX input conventions

- Input text is ASCII (or 8-bit codepage)

<table>
<thead>
<tr>
<th>Source text</th>
<th>Typeset output</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>'{a}</td>
<td>á</td>
<td>typical accent command</td>
</tr>
<tr>
<td>\c{c}</td>
<td>ç</td>
<td></td>
</tr>
<tr>
<td>\aa</td>
<td>å</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>—</td>
<td>ligature in typical \TeX fonts</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>α</td>
<td>math mode symbol</td>
</tr>
<tr>
<td>{\dn accha}</td>
<td>अच्छा</td>
<td>using custom preprocessor</td>
</tr>
</tbody>
</table>
Unicode support

Typesetting Unicode text with \texttt{Xe\TeX}

- Accented characters
  - many more than in any legacy codepage

\texttt{\textbackslash halign\{\textbackslash hfill\quad&
   \textbackslash hfill\cr
dan& dan\cr
dubok& dubok\cr
d\v{z}abe& d\v{z}ak\cr
d\v{z}in& d\v{z}abe\cr
D\v{z}in& d\v{z}in\cr
d\check{a}k& D\check{a}z\cr
Evropa& Evropa\cr\}
Typesetting Unicode text with \TeX

- CJK ideographs
  - they’re just more characters, no special effort required

\font\han="STSong" at 16pt
\font\rom="Gentium" at 8pt
\def\hc#1#2{\vtop{\hbox{\han #1}
  \hbox{\kern10pt\rom #2}}}
\vtop{\hc書く}{ka-ku}
\hc最も{motto-mo}
\hc最後{sai-go}
\hc働く{hatara-ku}
\hc海{umi}
Typesetting Unicode text with \TeX

- Complex scripts
  - just simple character data in the source file

\begin{verbatim}
\c 1
Shayadip jich aine
\p
\e nemz adax m tawursh 1
\v
Wey adiyi ke namse.
\v
Bie tertib nemz tcco na 2
\v
Denss yehnow yeeh narito
\v
We liyidi nass siednowow rjatam joo
\v
Adax nataam jie ketaap e
\v
Yee yiep ariq hor ji
\v
Yenshor \ldots\en mikh adox heezet 3
\v
Yee yip yenshor wos \ldots\yee
\end{verbatim}
Typesetting Unicode text with \TeX

- **Vertical text** (not fully supported)

\begin{verbatim}
\font\mon="Code2000:script=mong" at 18pt
\setbox0=\vbox{
  \hspace=3.6in \baselineskip=20pt
  \parindent=-12pt \leftskip=12pt
  \revpar \mon
  \par}
\special{x:gsave}\special{x:rotate -90}
\vskip-\ht0 \box0 \special{x:grestore}
\end{verbatim}
Unicode support

A cleaner multilingual solution

- All required characters directly represented
  - no need for “escape sequences” to access characters not included in the current codepage
  - no need to switch between codepages according to the language/script being typeset
  - characters rendered via standard access codes

- Character/glyph model and modern font rendering technologies
  - encoded text represents characters, not glyphs
  - complex script behavior separated from the encoded text data, handled through standard “smart font” technologies
Extending \TeX to process Unicode

Character codes

- Basic character codes are 16-bit
  - representing Unicode in the UTF-16 encoding form
  - (except when using legacy custom-encoded fonts)
- Extended \TeX primitives
  - `\char`, `\chardef` accept numbers up to 65,536
  - 4-digit hex notation using `^^^ab`c``
    \[ \texttt{\char"5609^^^6167} = 嘉慧 \]
- What about Unicode characters beyond Plane 0?
  - handled using surrogates (UTF-16 representation)
  - adequate for rendering
  - does not allow full per-character programmability
Extended \TeX code tables

- Per-character code tables \texttt{\catcode, \lccode, \uccode, \sfcode} enlarged
  - "plain \TeX" format initializes these tables based on Unicode character set
  - \texttt{\lowercase{DŽIN}}
    - džin
  - \texttt{\uppercase{Esi eyama klć míafe nuvćwo Ḟa vć la}}
    - ESI EYAMA KLĆ MÍAFE NUVĆWO ḞA VĆ LA
  - \texttt{\catcode`王=\active \def王{...}}
Input encodings

- By default, input read as Unicode (UTF-8 or UTF-16)
  - encoding form automatically detected
- Non-Unicode input text
  - legacy codepages supported via ICU converters
  - set codepage of current input file:
    \texttt{\LaTeXXinputencoding "charset-name"}
  - set initial codepage for newly-opened input files:
    \texttt{\LaTeXXdefaultencoding "charset-name"}
Extending \TeX\ to process Unicode

Hyphenation patterns

• Extended for 16-bit characters
• Standard hyphenation files are encoding-specific
  • modified to load correctly under \XeTeX
• Simple hyphenation for scripts such as Devanagari
  • text is simple character data, no macros, active chars, etc.

% break before or after any independent vowel
1अ1
1आ1
1इ1
% break after any dependent vowel, but never before
2ट1
2फ1
Font support in **Xe**TeX

**Host platform fonts**

- Use any font installed on the host computer
- \font command extended to accept “real” font names
- \font\rm="Trebuchet MS" at 16pt \rm Hello World!
  - Hello World!
- \font\it="Times Italic" at 16pt \it Hello World!
  - Hello World!
- \font\ch="Apple Chancery" at 16pt \ch Hello World!
  - Hello World!
- \font\heiti="STHeiti" at 16pt \heiti 你好，武汉!
  - 你好，武汉!
- No TFM files, etc., required to use new fonts!
Output device support

- Output driver uses the same fonts as the typesetting engine
  - no font name mapping files required
- Generate PDF as default output
  - there is actually an “extended DVI” (.xdv) intermediate
- Fonts automatically embedded and subsetted
Font support in \texttt{Xe\TeX}

Support for traditional \TeX{} fonts

- TFM files still supported
  - required for math fonts to provide precise metrics
  - implies non-Unicode data, using character codes 0...255 only
- PDF back-end supports Type 1 fonts
  - uses \texttt{.pfb} files in the \texttt{texmf} tree, just like \texttt{dvips}
  - no support for bitmap fonts
  - currently no \texttt{.vf} support
Font support in \TeX

Font mappings

- Traditional \TeX{} keyboarding practices
  - typical input:
    \``\LaTeX''---a typesetting system
  - generates: \``\LaTeX''---a typesetting system

- Font mapping for compatibility
  ; TECkit mapping for TeX input conventions
  U+002D U+002D \<\> U+2013 ; -- -> en dash
  U+002D U+002D U+002D \<\> U+2014 ; --- -> en dash
  U+0027 \<\> U+2019 ; ' -> right single quote
  U+0027 U+0027 \<\> U+201D ; '' -> right double quote
  U+0022 \> U+201D ; '' -> right double quote

- generates: \``\LaTeX''—a typesetting system
- the “font mapping” is associated with a specific \TeX{} font identifier
More fun with font mappings

\def\SampleText{Unicode - это уникальный код для любого символа, независимо от платформы, независимо от программы, независимо от языка.}
\font\gen="Gentium"
\gen\SampleText
\bigskip
\font\gentrans="Gentium: mapping=cyr-lat-iso9"
\gentrans\SampleText

Unicode - это уникальный код для любого символа, независимо от платформы, независимо от программы, независимо от языка.

Unicode - èto unikal'nyj kod dlâ lûbogo simvola, nezavisimo ot platformy, nezavisimo ot programmy, nezavisimo ot âzyka.
AAT font features

- Custom AAT features accessed via `\font` command

- `\font\x="Apple Chancery" at 16pt \x` The quick brown fox jumps over the lazy dog.
  - *The quick brown fox jumps over the lazy dog.*

- `\font\x="Apple Chancery:Letter Case=Small Caps;Design Complexity=Simple Design Level" at 16pt \x` The quick...
  - *THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.*

- `\font\x="Apple Chancery:Design Complexity=Flourishes Set A" at 16pt \x` The quick brown fox jumps over...
  - *The quick brown fox jumps over the lazy dog.*
Typographic features

OpenType: language and script

- Fonts may support multiple languages with differing behavior

```
\font\Doulos="Doulos SIL/ICU"
\font\DoulosViet="Doulos SIL/ICU:language=VIT"

Unicode cung cấp
một con số duy
nhất cho mỗi ký tự

\font\Brioso="Brioso Pro"
\font\BriosoTrk="Brioso Pro:language=TRK"

... gelen firmaları
... tarafından ...

... gelen firmaları
... tarafından ...
```
OpenType: language and script

- Complex Asian scripts require specific “shaping engines”
- With no “script tag”, only default Latin features applied

\font\x=\"Code2000\" \x \font\x=\"Code2000:script=arab\" \x \font\x=\"Code2000:script=deva\"

- Must load the font with the appropriate shaping engine
OpenType: optional features

- Font specification may include feature tags
  - \font\x="Brioso Pro" \x Hello World! 0123456789
    Hello World! 0123456789
  - \font\x="Brioso Pro:+smcp"
    HELLO WORLD! 0123456789
  - \font\x="Brioso Pro:+supers"
    Hello World! 0123456789
  - \font\x="Brioso Pro Italic:+onum"
    Hello World! 0123456789
  - \font\x="Brioso Pro Italic:+swsh,+zero"
    Hello World! Ø123456789
Typographic features

OpenType: optical sizing

- OpenType optical families automatically choose correct face for the size used
  - Brioso Pro at 7, 10, 18, 24pt sizes:
    - seven ten eighteen twenty-four

- Can override with /S= modifier on font name
  - showing different optical sizes using the same “at size”
    - Brioso Pro/S=7 Brioso Pro Caption
    - Brioso Pro/S=10 Brioso Pro Text
    - Brioso Pro/S=18 Brioso Pro Subhead
    - Brioso Pro/S=24 Brioso Pro Display
Asian-language linebreaking

Line-break positions

• Line breaking without word spaces
  • \TeX{} normally breaks lines at “glue” arising from spaces
  • Chinese, Japanese, Thai, etc. do not use word spaces
  • 基本上，计算机只是处理数字。它们指定一个数字，来储存字母或其他字符。在

• Use ICU line-break algorithm
  • find permitted line-break locations according to a specific locale
  • \texttt{\LaTeX}linebreaklocale "zh"
    基本上，计算机只是处理数字。它们指定一个数字，来储存字母或其他字符。在创造Unicode之前，有数百种指定这些数字的编码系统。没有一个编码可以包含足够的字符:
Justification

• Text without spaces is difficult for \TeX to justify
• Ragged-right setting is one solution

  • 基本上，计算机只是处理数字。它们指定一个数字，来储存字母或其他字符。在创造 Unicode 之前，有数百种指定这些数字的编码系统。没有一个编码可以包含足够的字符：

• Alternatively, use `\XeTeXlinebreakskip` to introduce glue at each potential break

  • 基本上，计算机只是处理数字。它们指定一个数字，来储存字母或其他字符。在创造 Unicode 之前，有数百种指定这些数字的编码系统。没有一个编码可以包含足够的字符：

• Could also use non-monospaced Latin characters

  • 基本上，计算机只是处理数字。它们指定一个数字，来储存字母或其他字符。在创造 Unicode 之前，有数百种指定这些数字的编码系统。没有一个编码可以包含足够的字符：
QuickTime image support

• Many graphic file formats directly supported
  • TIFF, JPEG, PNG, BMP, PICT, GIF, TGA, Photoshop, ...
  • \setbox0=\hbox{\TeXXpicfile "mypic.jpg"}

• Optional keywords to modify image
  • scaled, xscaled, yscaled, width, height, rotated

• Image width and height available to \TeX engine
• Can use via \LaTeX and \ConTeXt commands
Built-in graphics support

PDF documents

- Beware: QuickTime graphic importer accepts PDF
  - but renders as raster image at screen resolution!
- Use alternative command for true PDF inclusion
  - \texttt{\textbackslash Xe\TeX\textbackslash pdffile "xetex-intro-slides.pdf" page 1 scaled 400}

\textbf{Xe\TeX: the Multilingual Lion}
\textit{\TeX meets Unicode and smart fonts}

Jonathan Kew
SIL International
August 23, 2005
**fontspec.sty** by Will Robertson

- Simple specification of native OS X fonts in \texttt{\LaTeX}
- Integrates \texttt{Xe\LaTeX} font access with \texttt{\LaTeX} commands
  - setting the default document fonts
    \begin{verbatim}
    \usepackage{fontspec}
    \setromanfont{Adobe Garamond Pro}
    \setmonofont[Scale=0.8]{Andale Mono}
    \end{verbatim}
  - on-the-fly font and feature changes
    \begin{verbatim}
    Welcome to Wuhan,
    {\fontspec[LetterCase=SmallCaps]{China}}
    Welcome to Wuhan, \texttt{CHINA}
    August 25{\fontspec[VerticalPosition=Superior]{th}}
    August 25\textsuperscript{th}
    \end{verbatim}
xunicode.sty by Ross Moore

- Support for standard \LaTeX input of many special characters when using Unicode fonts
  - accent commands, named characters, etc., mapped to Unicode values for font access
  - does not handle dashes, quotes (use tex-text font mapping)
- Allows many non-Unicode \LaTeX documents to be processed using Unicode fonts
Using ConTeXt with XeTeX

- Reportedly works fairly readily, but not pre-configured “out of the box”
  - see http://www.contextgarden.net/XeTeX

- Use XeTeX font names and features in ConTeXt typescripts and other font definitions
  - see http://www.contextgarden.net/Fonts_in_XeTeX

\definedfont["Hoefler Text:
  mapping= tex-text;
  Style Options= Engraved Text;
  Letter Case= All Capitals;
  color= 229966" at 32pt]

Big Title
Status and future directions

What might be next for \texttt{Xe\TeX}?

- Ongoing bug-fixes and minor features
- Enhanced PDF back-end
  - leverage improved PDF support in Mac OS X 10.4
  - new \texttt{xdv2pdf} driver based on \texttt{dvipdfmx}
  - integration with \texttt{pdf\TeX} output routine
- True Unicode math support
  - requires extensions to \texttt{\textbackslash mathchar} etc., and underlying structures
  - also requires extended (at least 16-bit) font metric format
  - may be possible to make use of code from Ω
- \texttt{Xe\TeX} for non-Mac OS platforms
  - working towards integration with \TeX Live sources
Status and future directions

Questions... and answers?

• Contact information
  
  • mailto:jonathan_kew@sil.org

• XeTEX web site and mailing list
  
  • http://scripts.sil.org/xetex
  
  • http://tug.org/mailman/listinfo/xetex
  
  • svn://scripts.sil.org/xetex/TRUNK